

10030 LA SIMULACIÓN POR COMPUTADORA COMO ESTRATEGIA DIDÁCTICA PARA LA ENSEÑANZA DE LA COMPLEJIDAD ALGORÍTMICA

Brenda E. Meloni⁽¹⁾⁽²⁾, Leticia Constable⁽¹⁾⁽³⁾, Juan Giró⁽¹⁾⁽⁴⁾, Juan Carlos Vázquez⁽¹⁾⁽⁵⁾

⁽¹⁾Universidad Tecnológica Nacional

Facultad Regional Córdoba

Departamento de Ingeniería en Sistemas de Información

Cátedra de Sintaxis y Semántica de los Lenguajes

⁽²⁾bemeloni@gmail.com

⁽³⁾leticiaconstable@gmail.com

⁽⁴⁾juanfgiro@gmail.com

⁽⁵⁾jcjvazquez@gmail.com

Resumen: El proceso de enseñanza-aprendizaje del concepto de complejidad algorítmica, acarrea dificultades especiales debido al carácter abstracto de dicha noción. El uso de autómatas para el modelado de algoritmos, permite la aplicación de herramientas tales como simuladores de máquinas abstractas, a través de los cuales, se puede construir una imagen intuitiva del proceso al visualizar el comportamiento de dichos autómatas. Además los simuladores proveen información útil para el posterior cálculo de expresiones generales de la complejidad algorítmica. Se propone un ejercicio de aplicación integrador, a través del cual, con auxilio de un simulador de máquinas de Turing, se comparan las expresiones de la complejidad obtenidas para dos soluciones distintas de un mismo problema, empleando además conceptos previamente adquiridos en otras asignaturas, lo que favorece la transposición tanto horizontal como verticalmente respecto del diseño curricular de la carrera.

Palabras Clave: COMPLEJIDAD ALGORÍTMICA, MÁQUINAS DE TURING, SIMULADOR.

Introducción

La sociedad contemporánea se caracteriza por una gran diversidad y complejidad de escenarios y formas de comportamiento. Esta complejidad se ve aumentada por los cambios que afectan continuamente tanto la estructura de los escenarios, como las dinámicas sociales que en ellos ocurren.

En el ámbito académico, la experiencia de formación de los estudiantes universitarios debe prepararlos para enfrentar de manera adecuada tanto la complejidad e incertidumbre, como los cambios constantes, la necesidad de tomar decisiones de forma continua y de resolver de manera adecuada los problemas que se les presentan.

En este sentido, Edith Litwin [1] señala que “La enseñanza basada en casos, proyectos y/o problemas, así como la instrucción apoyada por las nuevas tecnologías de la información y la comunicación, constituyen herramientas pedagógicas que se apoyan en situaciones experienciales y que permiten tender un puente entre las situaciones de la vida real (la experiencia concreta con diversos aspectos de esa realidad), y los contextos de enseñanza y reflexión en el aula”.

La incorporación de la tecnología a los procesos de enseñanza superior requiere de transformaciones, ya que de nada sirve introducir nuevas tecnologías si no se producen cambios en el sistema de enseñanza. Además, toda propuesta didáctica se inserta y desarrolla en un tiempo y un espacio y quienes las diseñan tienen la responsabilidad de contemplar ambas dimensiones.

Una propuesta curricular estática que se apoye únicamente en metodologías de enseñanza transmisivo-receptivas difícilmente podrá enfrentar los retos que plantea la sociedad en continuo cambio. Por contraparte, una visión del currículum entendido dinámicamente deberá considerar y anticipar las condiciones en que los estudiantes enfrentarán el ejercicio profesional futuro y prever los requerimientos que esta prospectiva impone a la formación universitaria.

La relevancia del estudio de los temas de computabilidad y complejidad de algoritmos en las ingenierías, está estrechamente relacionada con la función básica de resolución de problemas de todo ingeniero y se refuerza en el caso de los ingenieros en sistemas, debido al enfoque lógico-matemático de las soluciones que éstos proponen, implementando algoritmos a través de programas de computación.

En la década de 1930, Gödel [2] primero, con su teorema de incompletitud de la aritmética, y Turing [3], Church [4] y otros luego, determinaron que los problemas a tratar pueden clasificarse en resolubles e irresolubles, según exista o no un algoritmo que los resuelva, es decir una secuencia ordenada de pasos que permita siempre arribar a un resultado en tiempo finito.

Los problemas resolubles por algoritmos también suelen denominarse “computables” y son los que presentan interés en la actividad profesional del ingeniero.

En el campo de los problemas computables, se presenta un nuevo desafío que consiste en determinar su complejidad, pero en abstracto, éste es un concepto esquivo (cantidad de variables involucradas, número de objetos y conceptos involucrados, grados de libertad, esfuerzo para entenderlo, etc.) y subjetivo, que varía según la persona que intenta resolverlo, sus conocimientos, su entrenamiento y destreza, y hasta con qué herramientas lo ataque. Por ello, este estudio se lleva a cabo analizando la complejidad de la *solución* diseñada para el mismo; como se sabe en algorítmica, siempre hay muchas soluciones para un mismo problema, por lo que se supone el análisis de *la mejor solución*. Esto último acarrea una dificultad no menor, en relación al teorema del aumento de velocidad de Blum [5], publicado en la década de 1960: la imposibilidad de determinar para muchos problemas, si la solución planteada “es” la mejor solución.

En el análisis de la complejidad [6] de la solución, se puede distinguir entre la complejidad directa o intrínseca, que para cuantificarla admite diversas métricas tales como puntos de función, líneas de código, complejidad estructural y ciclomática, por citar sólo algunas, que se discuten en varias asignaturas de las carreras de informática, y la complejidad operativa, que es una forma de medición indirecta cuyas métricas se refieren a los recursos (tiempo y espacio) requeridos para la ejecución del algoritmo de resolución diseñado, que se aborda en Sintaxis y Semántica de los Lenguajes (SSL), asignatura de la carrera Ingeniería en Sistemas de Información en la Universidad Tecnológica Nacional, Facultad Regional Córdoba.

Además, y en concordancia con los contenidos de SSL que entre otros temas trata las máquinas abstractas (que modelan algoritmos), se presenta el análisis de la complejidad indirecta con un enfoque mecanicista basado en Máquinas de Turing (ver figura 1).

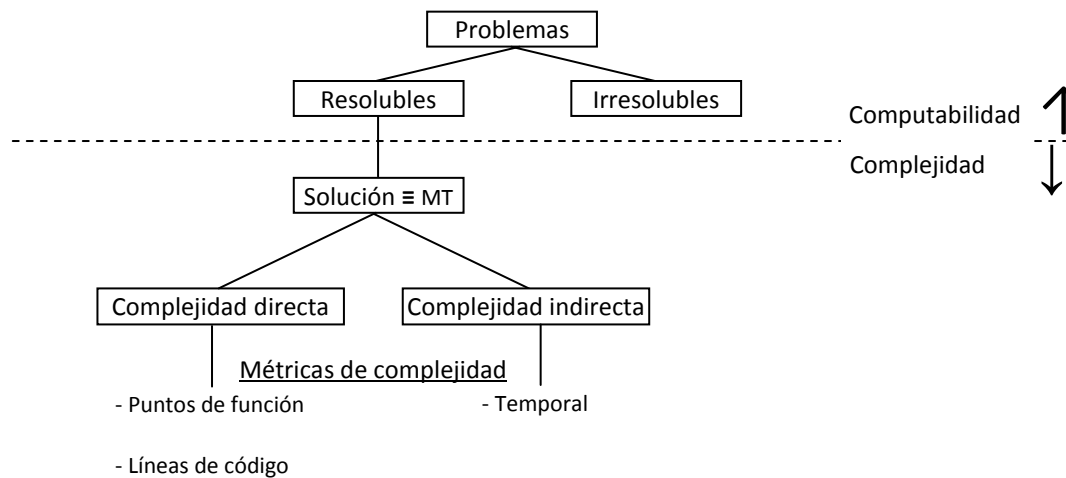


Figura 1: Marco conceptual de la computabilidad y la complejidad de problemas.

Por último, atendiendo a la ya mencionada imposibilidad de determinar que la solución bajo estudio es la mejor, la determinación de su complejidad en forma independiente pierde relevancia para dar lugar a la importancia de comparar las complejidades de distintas soluciones dadas a un mismo problema y así seleccionar la mejor de ellas.

Se presenta entonces a los estudiantes un problema, se construyen dos o más autómatas que modelan distintas soluciones válidas, se analizan las complejidades temporales y espaciales de cada solución y se comparan los resultados para determinar cuál es la mejor de las soluciones propuestas, es decir cuál es el mejor de los algoritmos disponibles.

Para la construcción de máquinas abstractas y la verificación de su comportamiento existen distintos simuladores y, algunos de ellos, presentan los datos necesarios para determinar las complejidades temporal y espacial.

La simulación es una herramienta que permite diseñar y desarrollar un modelo computarizado de un sistema o proceso, y realizar ensayos del modelo con el propósito de entender el comportamiento del sistema del mundo real o evaluar varias condiciones con las cuales puedan operar el sistema [7]. Está comprobado que la simulación es una técnica que puede ser utilizada como un instrumento pedagógico para enseñar a estudiantes habilidades básicas o para experimentar con nuevas situaciones, sobre las cuales se tiene poca o ninguna información, entre otras ventajas [8].

Desarrollo

El enfoque del estudio de la computabilidad que presentó Alan Turing en 1936, fue el de modelar el proceso de cálculo efectuado por un calculista humano, usando un esquema de máquina (abstracta) que rescataba los elementos y operaciones básicas esenciales involucradas en el mismo (sin tener en cuenta la intención de por qué lo hace).

Seguramente influenciado por los trabajos de Ramón y Cajal de 1905, quien descubrió que el sistema nervioso de los humanos se conforma de elementos simples y discretos interconectados (*doctrina de la neurona*), Turing supuso que el cerebro del calculista podía estar en cada instante, en uno de un número finito de estados posibles (muchos, muchísimos, pero finitos al fin).

Al leer un símbolo de su hoja de trabajo, el *estado cerebral* del calculista cambiará y podrá entonces eventualmente modificar el símbolo leído, para luego leer el siguiente o el anterior. Este proceso se repetirá hasta dar por terminado el cálculo.

Turing también especuló que el conjunto de símbolos (alfabeto) que el calculista maneja debería ser finito, ya que no podría distinguir infinitos símbolos.

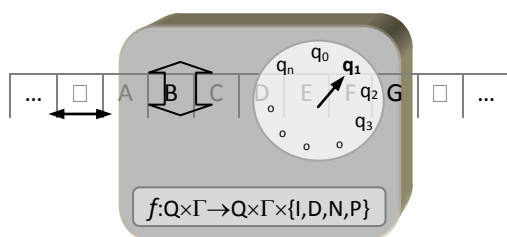


Figura 2: Máquina de Turing.

La figura 2 muestra un esquema de una Máquina de Turing, en la cual el conjunto finito de estados posibles Q está indicado en el círculo (la flecha marca el estado actual), la hoja de papel está representada por una cinta infinita dividida en celdas, cada una de las cuales puede contener un solo símbolo de algún alfabeto de cinta Γ (que incluye al alfabeto de entrada con el cual pueden escribirse las cadenas iniciales a procesar por la máquina, un símbolo especial de espacio en blanco que rellena el resto de la cinta infinita y algunos símbolos auxiliares necesarios para el proceso), y el funcionamiento de la misma está regido por una función f (denominada su *función de transición*) que para cada par estado-símbolo indica el nuevo estado al que se transitará, lo que se grabará en la cinta y la acción mecánica que deberá realizarse: mover el cabezal de lecto-escritura a la (L)zquierda, a la (D)erecha o (N)o moverlo, o eventualmente detener el cálculo (P).

Matemáticamente, este modelo se describe con una séptupla $(\Sigma, \Gamma, Q, q_0, A, f, \square)$ que agrupa al alfabeto de entrada, alfabeto de cinta, conjunto de estados, estado inicial, conjunto de estados de aceptación (a los que se arriba al final del proceso si la cadena de entrada es reconocida), función de transición y símbolo especial de espacio en blanco.

Esta máquina puede pensarse como una *máquina reconocedora*, que aceptará la cadena de entrada si al procesarla con la función f logra detenerse en un estado de aceptación (la rechaza en caso contrario), o como una *máquina traductora-ejecutora de procedimientos*, que toma una cadena de entrada y al detenerse deja en su cinta el resultado de la operación, como salida.

Este dispositivo abstracto de cómputo, es ideal para el estudio de los conceptos de computabilidad y complejidad.

La llamada *tesis de Turing-Church* establece que un problema será considerado computable si y sólo si, puede diseñarse una máquina de Turing que lo resuelva. Por otro lado, para determinar la complejidad operativa de la solución, se define la *complejidad espacial* de la máquina $E(n)$ como la cantidad de celdas de cinta usadas para procesar una cadena de entrada en función de su largo n , y la *complejidad temporal* $T(n)$ como la relación que existe entre el largo n de la cadena de entrada a procesar y el tiempo demandado en la tarea, medido como la cantidad de transiciones efectuadas para procesarla. Decimos que es ideal porque representa un dispositivo abstracto bien definido, cuyas métricas de complejidad resultan ser independientes tanto de la tecnología de implantación como de los usuarios que la utilicen.

Se detallará a continuación, y a modo de ejemplo, un ejercicio del tipo de los que se proponen a los estudiantes de la asignatura SSL en la UTN-FRC, y su resolución. [5]

Ejercicio:

Dado un lenguaje de palíndromos impares tal como $L = \{\alpha\#\alpha^{-1} \mid \alpha \in \{a, b\}^+\}$ se pide:

- Construya una máquina de Turing (MT) que acepte las cadenas de dicho lenguaje.
- Construya otra máquina de Turing que se comporte como un autómata con pila, como solución alternativa, para realizar el mismo trabajo (parte de la cinta deberá ser usada con acceso LIFO para simular el trabajo de una pila).
- Verifique la validez de las máquinas construidas en los puntos a) y b) con el auxilio de un simulador de máquinas de Turing, probando los modelos de solución con distintas cadenas.
- Obtenga expresiones generales para la complejidad temporal y espacial de ambas máquinas en forma analítica y gráfica.
- Compare las complejidades obtenidas para los dos procedimientos de solución y exprese a qué conclusión se puede arribar.

Solución:

a) Para resolver este problema se propone una MT que mueve el cabezal hasta el separador $\#$, y a partir de allí avanza desde el medio de la cadena hacia los extremos, verificando la concordancia de símbolos del sufijo y del prefijo, con sucesivos movimientos de avance y retroceso.

La definición formal de esta máquina es: $MT_1 = (\{a, b, \#\}, \{a, b, \#, 1, 2, \square\}, \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, q_0, \{q_6\}, f, \square)$ y su función de transición f puede verse representada en el grafo de la figura 3.

b) Para la segunda solución, se construye una MT que mueve el cabezal hasta el final de la cadena y escribe un símbolo P en el primer blanco para marcar el inicio de la pila, luego el cabezal vuelve al inicio de la cadena y por cada símbolo que lee del prefijo, hasta el separador $\#$, lo escribe en la pila. A continuación, por cada símbolo del

sufijo posterior al separador #, si el caracter leído en la cadena coincide con el de la cima de la pila (el más a la derecha en la cinta), lo marca como procesado y lo desapila (reemplazando con un blanco el símbolo de más a la derecha de la cinta).

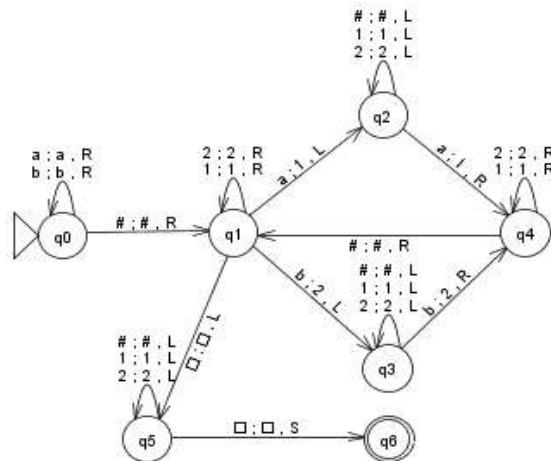


Figura 3: MT_1 que avanza desde el medio de la cadena hacia los extremos.

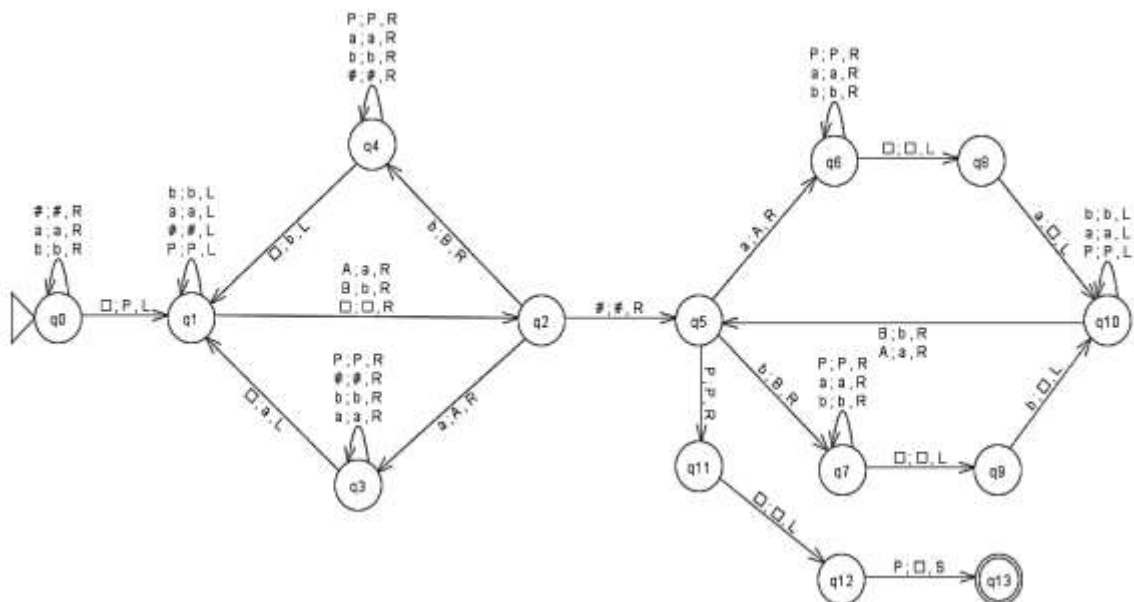


Figura 4: MT_2 que utiliza parte de su cinta como una pila.

La definición formal de la segunda máquina es: $MT_2 = (\{a, b, \#\}, \{a, b, \#, A, B, P, \square\}, \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}\}, q_0, \{q_{13}\}, f, \square)$ donde la función de transición f se representa a través del grafo de la figura 4.

d) Para obtener analíticamente la expresión de la complejidad temporal, se determina el total de intervalos de tiempo (esto es, transiciones) necesarios para procesar las cadenas de ciertos largos (por ejemplo para $n = 3, 5, 7$ y 9), en cada máquina. Como este recuento puede ser bastante laborioso, suele recurrirse a los simuladores para la tarea, ya sea efectuando la simulación paso a paso, o tomando los totales arrojados por el simulador, si es que éste los ofrece. De manera análoga, para el caso de la complejidad espacial, se determina el total de celdas de cinta necesarias para validar las mismas cadenas. Estos valores, para el problema y las máquinas planteadas son presentados en la tabla 1.

Observando la variación de los valores obtenidos para tiempos y espacios en ambas máquinas, se puede inferir que la complejidad temporal es de mayor orden que la espacial (crecen más rápido).

n	MT_1		MT_2	
	$T(n)$	$E(n)$	$T(n)$	$E(n)$
3	11	5	28	6
5	23	7	60	9
7	39	9	104	12
9	59	11	160	15

Tabla 1: Variación de las complejidades en función de la longitud de la cadena de entrada

Se trata a continuación de proponer expresiones generales de las complejidades temporales. Para ello, por inspección de los valores que figuran en la tabla 1, se asume inicialmente, que se trata de complejidades de tipo polinómico y que dichos polinomios son de grado no superior a tres:

$$T(n) = An^3 + Bn^2 + Cn + D$$

Se calculan los coeficientes de los polinomios que responden a estos valores tabulados, recurriendo para ello al álgebra matricial, y se plantean los siguientes sistemas de ecuaciones para cada máquina:

Para MT_1

$$\begin{bmatrix} 3^3 & 3^2 & 3 & 1 \\ 5^3 & 5^2 & 5 & 1 \\ 7^3 & 7^2 & 7 & 1 \\ 9^3 & 9^2 & 9 & 1 \end{bmatrix} \begin{Bmatrix} A \\ B \\ C \\ D \end{Bmatrix} = \begin{Bmatrix} 11 \\ 23 \\ 39 \\ 59 \end{Bmatrix}$$

Para MT_2

$$\begin{bmatrix} 3^3 & 3^2 & 3 & 1 \\ 5^3 & 5^2 & 5 & 1 \\ 7^3 & 7^2 & 7 & 1 \\ 9^3 & 9^2 & 9 & 1 \end{bmatrix} \begin{Bmatrix} A \\ B \\ C \\ D \end{Bmatrix} = \begin{Bmatrix} 28 \\ 60 \\ 104 \\ 160 \end{Bmatrix}$$

Las soluciones de estos sistemas son:

Para MT_1

$$\begin{cases} A \\ B \\ C \\ D \end{cases} = \begin{cases} 0,00 \\ 0,50 \\ 2,00 \\ 0,50 \end{cases}$$

Para MT_2

$$\begin{cases} A \\ B \\ C \\ D \end{cases} = \begin{cases} 0,00 \\ 1,50 \\ 4,00 \\ 2,50 \end{cases}$$

La anulación del coeficiente **A**, muestra que la complejidad temporal está correctamente representada por un polinomio de segundo grado, es decir de $O(n^2)$. Así, a partir de los valores de los coeficientes **B**, **C** y **D** se puede disponer de las expresiones para las complejidades temporales:

Para MT_1
$$T_1(n) = \frac{1}{2}n^2 + 2n + \frac{1}{2} \quad (1)$$

Para MT_2
$$T_2(n) = \frac{3}{2}n^2 + 4n + \frac{5}{2} \quad (2)$$

Para la complejidad espacial, en el caso de MT_1 , por simple inspección (aunque podría intentarse una demostración por inducción matemática) de los valores de la tabla 1, se puede deducir que responde a la expresión:

$$E_1(n) = n + 2 \quad (3)$$

En el caso de MT_2 , se sigue para la determinación de la complejidad espacial el mismo procedimiento usado para el cálculo de expresiones de la complejidad temporal: por inspección de los valores de la tabla 1, se propone un polinomio que se asume de grado no superior a dos.

$$E_2(n) = An^2 + Bn + C$$

Se puede plantear entonces el sistema de ecuaciones:

$$\begin{bmatrix} 3^2 & 3 & 1 \\ 5^2 & 5 & 1 \\ 7^2 & 7 & 1 \end{bmatrix} \begin{Bmatrix} A \\ B \\ C \end{Bmatrix} = \begin{Bmatrix} 6 \\ 9 \\ 12 \end{Bmatrix}$$

cuya solución es:

$$\begin{Bmatrix} A \\ B \\ C \end{Bmatrix} = \begin{Bmatrix} 0,00 \\ 1,50 \\ 1,50 \end{Bmatrix}$$

Luego, la complejidad espacial para MT_2 resulta:

$$E_2(n) = \frac{3}{2}n + \frac{3}{2} \quad (4)$$

El valor nulo del coeficiente **A** significa en este caso, que la complejidad espacial es representada por un polinomio de primer grado, es decir $O(n)$.

Una alternativa al procedimiento seguido, sería la de obtener por medios gráficos las expresiones de las complejidades tanto temporales como espaciales, para ambas soluciones; se puede utilizar para esto una planilla de cálculo como herramienta, para representar gráficamente los valores volcados de la tabla 1 y aproximar las curvas con polinomios de diversos grados, hasta obtener representaciones no oscilantes solicitando las fórmulas de las curvas de tendencia.

Con este recurso gráfico se obtienen las curvas y expresiones de las figuras 5 y 6.

Nótese que las expresiones obtenidas deben siempre ser utilizadas para valores de n que estén próximos al intervalo de los casos conocidos, aquí fueron $3 \leq n \leq 9$. Nunca es conveniente hacer predicciones para un valor muy lejano del intervalo simulado (por ejemplo $n > 100$) porque podría ocurrir que en la zona conocida haya varias funciones que representen correctamente los puntos, pero que luego divergen cuando n crece. En ese caso la predicción puede arrojar resultados erróneos.

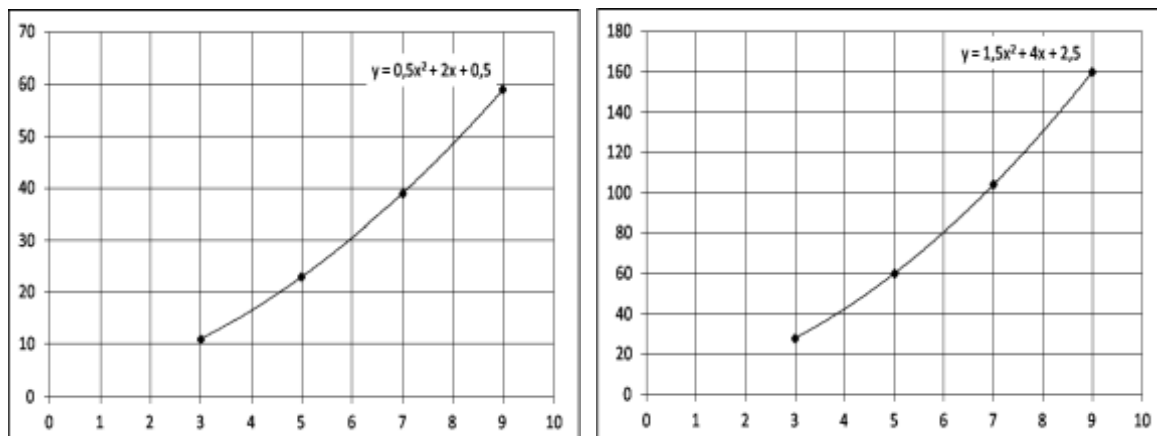


Figura 5: Obtención gráfica de la complejidad temporal de MT_1 a la izquierda y MT_2 a la derecha.

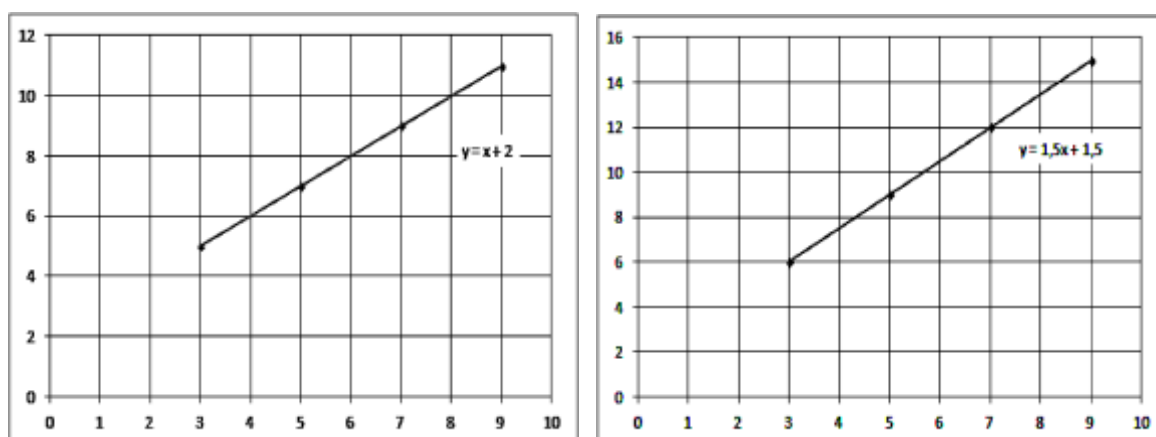


Figura 6: Obtención gráfica de la complejidad espacial de MT_1 a la izquierda y MT_2 a la derecha.

Por otra parte, resulta muy conveniente proponer un polinomio de grado mayor al que suponemos que es el “verdadero” grado de la expresión. Si el grado propuesto resultara excesivo, como en el caso que se presentó, los coeficientes de los términos de mayor grado al necesario resultarán nulos, pero contrariamente, si se propone inicialmente un polinomio de menor grado que el óptimo, la función de aproximación será válida en un intervalo acotado, lo que no será evidente y puede llevar a la utilización indebida de la fórmula de complejidad obtenida, brindando resultados erróneos.

Como parte del proyecto de investigación, se desarrolló un simulador que incluye la funcionalidad de calcular la expresión polinómica de la complejidad algorítmica de una máquina de Turing.

Las figuras 7 y 8 muestran los resultados de las ejecuciones de MT_1 , con cadenas de longitud 3, 5, 7 y 9 caracteres, tales como las que figuran en la Tabla 1.

Finalmente, luego de estas cuatro ejecuciones, el simulador presenta la expresión polinómica resultante de la complejidad calculada. Esta expresión sirve al alumno como verificación de resultado.



Figura 7: Resultados de la ejecución de MT_1 con el simulador para cadenas de longitud 3 y 5



Figura 8: Resultados de la ejecución de MT_1 con el simulador para cadenas de longitud 7 y 9

Discusión de resultados y trabajo futuro

Desde el punto de vista del aprendizaje se percibe una buena motivación por parte de los estudiantes y una excelente captación conceptual del tema; el hecho de utilizar la simulación para verificar el correcto funcionamiento de la MT y poder simular con distintas cadenas de entrada su comportamiento, resultó un complemento interesante a la enseñanza y el aprendizaje significativo del tema; además, el hecho de contar con simuladores que calculan la complejidad espacial y temporal, facilita el análisis. En particular, en referencia al simulador desarrollado, se seguirán ampliando sus funcionalidades para permitir el cálculo de complejidades polinómicas de mayor grado y aún de tipo exponencial.

Las evaluaciones de los informes escritos y de las defensas orales que se solicitaron a los estudiantes, han resultado altamente satisfactorias.

Con esta base se piensa que el enfoque mecanicista seleccionado y el uso de simuladores, tuvo resultados positivos en el proceso de enseñanza-aprendizaje.

Las ventajas de incluir las TICs en el proceso educativo ya no se encuentran en discusión. A través de ellas los alumnos, enriquecen su aprendizaje con elementos audiovisuales, manejan sus tiempos de aprendizaje independizándose de los horarios prefijados de clase por el sistema tradicional y disponen de herramientas de cálculo, de consulta y de graficación entre otras, que agilizan sus trabajos, por citar algunas ventajas. Parece particularmente válido este enfoque en el caso que nos ocupa, ya que son estudiantes de Ingeniería en Sistemas de Información lo que presupone

interés por el uso de software y computadoras. Esto no debe resultar en docentes que deleguen o resignen su tarea, sino que deben estar alertas, acompañando y guiando a los estudiantes para evacuar dudas durante todo el camino, para que logren la transferencia de conocimientos relacionados de otras asignaturas como por ejemplo, análisis, variación, aproximación, acotación y convergencia de funciones, resolución de sistemas de ecuaciones, manejo de herramientas de software, simulación, etcétera, y por último, para contribuir al desarrollo de la capacidad de análisis de los resultados obtenidos.

Conclusión

Para la enseñanza del difícil concepto de complejidad de los problemas, en SSL se ha optado por el análisis de la complejidad de las soluciones modeladas con máquinas de Turing a esos problemas (en esencia, soluciones algorítmicas).

Si bien se tratan introductoriamente los distintos tipos de métricas y de enfoques que pueden darse de la complejidad de problemas, se hace foco en la complejidad operativa que usualmente es utilizada en ciencias de la computación, definiendo y calculando tanto el tiempo como el espacio requeridos por la máquina para solucionar el problema.

Para la fijación de conceptos y la necesaria habilidad práctica para resolver problemas y evaluar las soluciones propuestas, que es competencia de los futuros Ingenieros en Sistemas de Información, se plantea un práctico integrador que, además de hacer efectiva la práctica de los temas teóricos sobre autómatas y lenguajes impartidos en SSL, necesita para su desarrollo el uso de conocimientos de varias asignaturas de la carrera (especificación de requerimientos, valoración de los mismos para distintos simuladores posibles, diseño de soluciones, evaluación de la bondad de las mismas, manejo de funciones del análisis matemático, resolución de sistemas de ecuaciones del álgebra, etc.).

Gracias a esta práctica, se ha evidenciado una mejora en la comprensión por parte de los estudiantes de los temas tratados en SSL y, mediante encuestas informales realizadas a los estudiantes, se pudo constatar la conformidad y beneplácito con la actividad.

Referencias

- [1] LITWIN, Edith (1997). "Las configuraciones didácticas. Una nueva agenda para la enseñanza superior". Editorial: Paidós Ibérica. España.
- [2] MELTZER, B. and BRAITHWAITE, R. B. (1962). "On Formally Undecidable Propositions of Principia Mathematica and Related System", Dover Publications, (Dover Edition 1992). EEUU.
- [3] ANDERSON, Anthony and ZELNY, Michel (2001). "Logic, Meaning and Computación. Essays en memory of Alonzo Church". Published by Kluwer Academic Publishers.
- [4] STRATHER, Paul (1999). "Turing y el ordenador". Siglo Veintiuno de España Editores. S.A. Madrid, España.

[5] CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald L. & STEIN, Clifford, (2010). "Introduction to Algorithms", 3ra edición, MIT Press and McGraw-Hill.

[6] GIRÓ, Juan; VÁZQUEZ, Juan; MELONI, Brenda; CONSTABLE, Leticia (2015). "Lenguajes Formales y Teoría de Autómatas". Editorial Alfaomega. Bs. As. Argentina.

[7] SHANNON, R. (1988). "Simulación de sistemas: Diseño, desarrollo e implantación". Editorial Trillas, México.

[8] NAYLOR, Thomas H. (1993). "Técnicas de simulación en computadoras". Primera edición. Editorial Limusa. México.

Agradecimientos.

UTN 3591: *Evaluación del impacto de variantes no convencionales, en el desempeño de autómatas finitos con memoria de pila*, el proyecto de investigación en el cual se genera el presente artículo, y en donde se construyó el simulador ejemplificado, tiene financiamiento de la Facultad Córdoba y de la Secretaría de Ciencia, Tecnología y Posgrado de UTN.